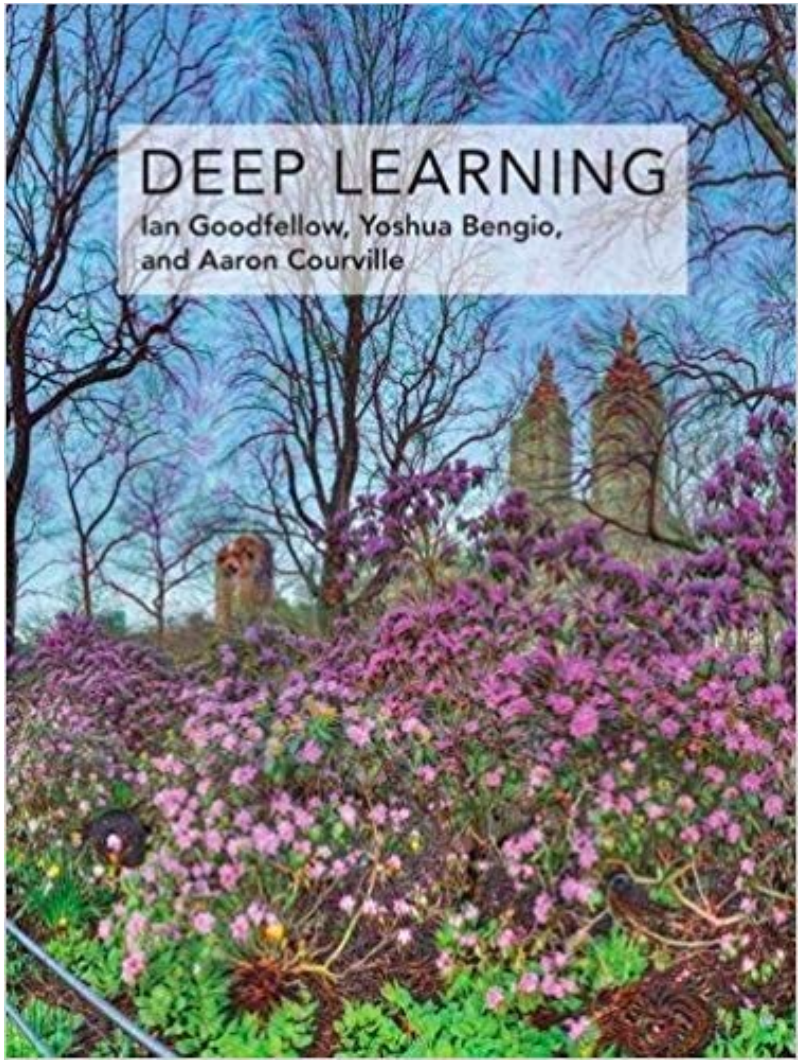
國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Deep Learning
深度學習
Fall 2025

Machine Learning Basics

(Chapter 5.4-5.7)

Prof. Chia-Han Lee
李佳翰 教授



- Figure source: Textbook and Internet
- You are encouraged to buy the textbook.
- Please respect the copyright of the textbook. Do not distribute the materials to other people.

Estimators, bias and variance

- The field of statistics gives us many tools to achieve the machine learning goal of solving a task not only on the training set but also to generalize.

- Foundational concepts such as parameter estimation, bias and variance are useful to formally characterize notions of generalization, underfitting and overfitting.

Point estimation

- Point estimation is the attempt to provide the single "best" prediction of some quantity of interest. In general the quantity of interest can be a single parameter or a vector of parameters in some parametric model, such as the weights in our linear regression example.

- To distinguish estimates of parameters from their true value, our convention will be to denote a point estimate of a parameter $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$.

- Let $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ be a set of m independent and identically distributed (i.i.d.) data points. A point estimator or statistic is any function of the data:

$$\hat{\boldsymbol{\theta}}_m = g(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}). \tag{5.19}$$

Point estimation

- While almost any function thus qualifies as an estimator, a good estimator is a function whose output is close to the true underlying $\boldsymbol{\theta}$ that generated the training data.

- For now, we take the frequentist perspective on statistics. That is, we assume that the true parameter value $\boldsymbol{\theta}$ is fixed but unknown, while the point estimate $\widehat{\boldsymbol{\theta}}$ is a function of the data.

- Since the data is drawn from a random process, any function of the data is random. Therefore $\widehat{\boldsymbol{\theta}}$ is a random variable.

Function estimation (point estimation)

- We are trying to predict a variable $\boldsymbol{y}$ given an input vector $\boldsymbol{x}$. We assume that there is a function $f(\boldsymbol{x})$ that describes the approximate relationship between $\boldsymbol{y}$ and $\boldsymbol{x}$. For example, assume that $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ stands for the part of $\boldsymbol{y}$ that is not predictable from $\boldsymbol{x}$.

- In function estimation, we are interested in approximating f with a model or estimate $\hat{f}$. The function estimator $\hat{f}$ is simply a point estimator in function space. The linear regression example and the polynomial regression example both illustrate scenarios that may be interpreted as either estimating a parameter $\boldsymbol{\omega}$ or estimating a function $\hat{f}$ mapping from $\boldsymbol{x}$ to y.

Bias

- The bias of an estimator is defined as

$$\text{bias}(\hat{\boldsymbol{\theta}}_m) = \mathbb{E}(\hat{\boldsymbol{\theta}}_m) - \boldsymbol{\theta}, \tag{5.20}$$

where the expectation is over the data (seen as samples from a random variable) and $\boldsymbol{\theta}$ is the true underlying value of $\boldsymbol{\theta}$ used to define the data-generating distribution.

- An estimator $\widehat{\boldsymbol{\theta}}_m$ is said to be unbiased if $\text{bias}(\widehat{\boldsymbol{\theta}}_m) = \mathbf{0}$, which implies that $\mathbb{E}(\widehat{\boldsymbol{\theta}}_m) = \boldsymbol{\theta}$. An estimator $\widehat{\boldsymbol{\theta}}_m$ is said to be asymptotically unbiased if $\lim_{m\to\infty} \text{bias}(\widehat{\boldsymbol{\theta}}_m) = \mathbf{0}$, which implies that $\lim_{m\to\infty} \mathbb{E}(\widehat{\boldsymbol{\theta}}_m) = \boldsymbol{\theta}$.

Bias

Example: Bernoulli Distribution

- Consider a set of samples $\{x^{(1)}, \ldots, x^{(m)}\}$ that are independently and identically distributed according to a Bernoulli distribution with mean θ:

$$P(x^{(i)}; \theta) = \theta^{x^{(i)}}(1-\theta)^{(1-x^{(i)})}. \tag{5.21}$$

- A common estimator for the θ parameter of this distribution is the mean of the training samples:

$$\hat{\theta}_m = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}. \tag{5.22}$$

Bias

- To determine whether this estimator is biased, we can substitute equation 5.22 into equation 5.20:

$$\text{bias}(\hat{\theta}_m) = \mathbb{E}[\hat{\theta}_m] - \theta \tag{5.23}$$

$$= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right] - \theta \tag{5.24}$$

$$= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\left[x^{(i)}\right] - \theta \tag{5.25}$$

$$= \frac{1}{m}\sum_{i=1}^{m} \sum_{x^{(i)}=0}^{1} \left(x^{(i)} \theta^{x^{(i)}} (1-\theta)^{(1-x^{(i)})}\right) - \theta \tag{5.26}$$

$$= \frac{1}{m}\sum_{i=1}^{m} (\theta) - \theta \tag{5.27}$$

$$= \theta - \theta = 0 \tag{5.28}$$

- Since $\text{bias}(\hat{\theta}) = 0$, we say that our estimator $\hat{\theta}$ is unbiased.

Bias

Example: Gaussian Distribution Estimator of the Mean

- Consider a set of samples $\{x^{(1)}, \ldots, x^{(m)}\}$ that are independently and identically distributed according to a Gaussian distribution $p(x^{(i)}) = \mathcal{N}(x^{(i)}; \mu, \sigma^2)$, where $i \in \{1, \ldots, m\}$. Recall that the Gaussian probability density function is given by

$$p(x^{(i)}; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x^{(i)} - \mu)^2}{\sigma^2} \right). \tag{5.29}$$

- A common estimator of the Gaussian mean parameter is known as the sample mean:

$$\hat{\mu}_m = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \tag{5.30}$$

Bias

- To determine the bias of the sample mean, we are again interested in calculating its expectation:

$$\text{bias}(\hat{\mu}_m) = \mathbb{E}[\hat{\mu}_m] - \mu \tag{5.31}$$

$$= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right] - \mu \tag{5.32}$$

$$= \left(\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\left[x^{(i)}\right]\right) - \mu \tag{5.33}$$

$$= \left(\frac{1}{m}\sum_{i=1}^{m} \mu\right) - \mu \tag{5.34}$$

$$= \mu - \mu = 0 \tag{5.35}$$

- Thus we find that the sample mean is an unbiased estimator of Gaussian mean parameter.

Bias

Example: Estimators of the Variance of a Gaussian Distribution

- For this example, we compare two different estimators of the variance parameter σ^2 of a Gaussian distribution. We are interested in knowing if either estimator is biased.

- The first estimator of σ^2 we consider is known as the sample variance

$$\hat{\sigma}_m^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \hat{\mu}_m\right)^2, \tag{5.36}$$

where $\hat{\mu}_m$ is the sample mean. More formally, we are interested in computing

$$\text{bias}(\hat{\sigma}_m^2) = \mathbb{E}[\hat{\sigma}_m^2] - \sigma^2. \tag{5.37}$$

Bias

- We begin by evaluating the term $\mathbb{E}[\hat{\sigma}_m^2]$:

$$\mathbb{E}[\hat{\sigma}_m^2] = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \hat{\mu}_m\right)^2\right] \tag{5.38}$$

$$= \frac{m-1}{m}\sigma^2 \tag{5.39}$$

- Returning to equation 5.37, we conclude that the bias of $\hat{\sigma}_m^2$ is $-\sigma^2/m$. Therefore, the sample variance is a biased estimator.

Bias

- The unbiased sample variance estimator

$$\tilde{\sigma}_m^2 = \frac{1}{m-1}\sum_{i=1}^{m}\left(x^{(i)} - \hat{\mu}_m\right)^2 \tag{5.40}$$

provides an alternative approach. As the name suggests this estimator is unbiased. That is, we find that $\mathbb{E}[\tilde{\sigma}_m^2] =$

$$\begin{aligned}
\mathbb{E}[\tilde{\sigma}_m^2] &= \mathbb{E}\left[\frac{1}{m-1}\sum_{i=1}^{m}\left(x^{(i)} - \hat{\mu}_m\right)^2\right] && (5.41)\\
&= \frac{m}{m-1}\mathbb{E}[\hat{\sigma}_m^2] && (5.42)\\
&= \frac{m}{m-1}\left(\frac{m-1}{m}\sigma^2\right) && (5.43)\\
&= \sigma^2. && (5.44)
\end{aligned}$$

- We have two estimators: one is biased, and the other is not. While unbiased estimators are clearly desirable, they are not always the “best” estimators. We often use biased estimators that possess important properties.

Variance and standard error

- Another property of the estimator that we might want to consider is how much we expect it to vary as a function of the data sample.

- The variance of an estimator is simply the variance $\mathrm{Var}(\hat{\theta})$, where the random variable is the training set. Alternately, the square root of the variance is called the standard error, denoted $\mathrm{SE}(\hat{\theta})$.

- The variance, or the standard error, of an estimator provides a measure of how we would expect the estimate we compute from data to vary as we independently resample the dataset from the underlying data-generating process. We would like it to have relatively low variance.

Variance and standard error

- The standard error of the mean is given by

$$\mathrm{SE}(\hat{\mu}_m) = \sqrt{\mathrm{Var}\left[\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right]} = \frac{\sigma}{\sqrt{m}}, \qquad (5.46)$$

  where σ^2 is the true variance of the samples $x^{(i)}$.

- The standard error is often estimated by using an estimate of σ. Unfortunately, neither the square root of the sample variance nor the square root of the unbiased estimator of the variance provide an unbiased estimate of the standard deviation. Both approaches tend to underestimate the true standard deviation but are still used in practice. The square root of the unbiased estimator of the variance is less of an underestimate.

- For large m, the approximation is quite reasonable.

Variance and standard error

- We often estimate the generalization error by computing the sample mean of the error on the test set. The number of examples in the test set determines the accuracy.

- Taking advantage of the central limit theorem, which tells us that the mean will be approximately distributed with a normal distribution, we can use the standard error to compute the probability that the true expectation falls in any chosen interval. For example, the 95 percent confidence interval centered on the mean $\hat{\mu}_m$ is

$$(\hat{\mu}_m - 1.96\mathrm{SE}(\hat{\mu}_m), \hat{\mu}_m + 1.96\mathrm{SE}(\hat{\mu}_m)), \tag{5.47}$$

under the normal distribution with mean $\hat{\mu}_m$ and variance $\mathrm{SE}(\hat{\mu}_m)^2$.

Variance and standard error

Example: Bernoulli Distribution

- We once again consider a set of samples $\{x^{(1)}, \ldots, x^{(m)}\}$ drawn independently and identically from a Bernoulli distribution (recall $P(x^{(i)}; \theta) = \theta^{x^{(i)}}(1 - \theta)^{(1-x^{(i)})}$).

- This time we are interested in computing the variance of the estimator $\hat{\theta}_m = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$.

Variance and standard error

$$\text{Var}\left(\hat{\theta}_m\right) = \text{Var}\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)}\right) \tag{5.48}$$

$$= \frac{1}{m^2}\sum_{i=1}^{m} \text{Var}\left(x^{(i)}\right) \tag{5.49}$$

$$= \frac{1}{m^2}\sum_{i=1}^{m} \theta(1-\theta) \tag{5.50}$$

$$= \frac{1}{m^2} m\theta(1-\theta) \tag{5.51}$$

$$= \frac{1}{m}\theta(1-\theta) \tag{5.52}$$

- The variance of the estimator decreases as a function of m, the number of examples in the dataset. This is a common property of popular estimators.

Trading off bias and variance

- Bias measures the expected deviation from the true value of the function or parameter. Variance provides a measure of the deviation from the expected estimator value that any particular sampling of the data is likely to cause.

- What happens when we are given a choice between two estimators, one with more bias and one with more variance? How do we choose between them?

Trading off bias and variance

- The most common way to negotiate this trade-off is to use cross-validation. Alternatively, we can also compare the mean squared error (MSE) of the estimates:

$$\text{MSE} = \mathbb{E}[(\hat{\theta}_m - \theta)^2] \tag{5.53}$$

$$= \text{Bias}(\hat{\theta}_m)^2 + \text{Var}(\hat{\theta}_m) \tag{5.54}$$

- The MSE measures the overall expected deviation—in a squared error sense— between the estimator and the true value of the parameter θ. Evaluating the MSE incorporates both the bias and the variance.

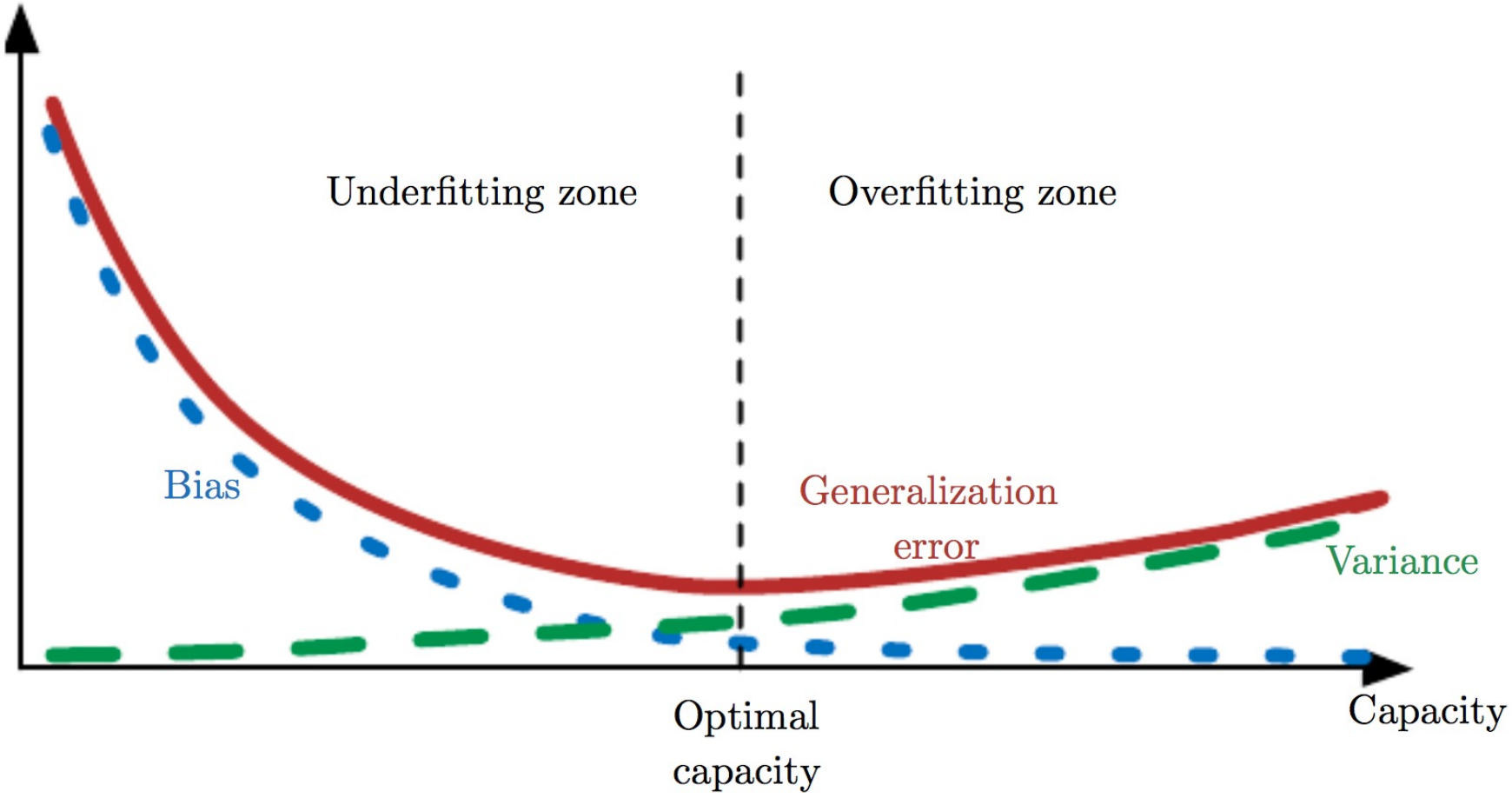
- Desirable estimators are those with small MSE and these are estimators that manage to keep both their bias and variance somewhat in check.

Trading off bias and variance

- The relationship between bias and variance is tightly linked to the machine learning concepts of capacity, underfitting and overfitting.

- When generalization error is measured by the MSE (where bias and variance are meaningful components of generalization error), increasing capacity tends to increase variance and decrease bias. We see the U-shaped curve of generalization error as a function of capacity.

Trading off bias and variance



Maximum likelihood estimation

- Rather than guessing that some function might make a good estimator and then analyzing its bias and variance, we would like to have some principle from which we can derive specific functions that are good estimators for different models.

- The most common such principle is the maximum likelihood principle.

Maximum likelihood estimation

- Consider a set of m examples $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ drawn independently from the true but unknown data-generating distribution $p_{\text{data}}(\boldsymbol{x})$.

- Let $p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})$ be a parametric family of probability distributions over the same space indexed by $\boldsymbol{\theta}$. In other words, $p_{\text{model}}(\boldsymbol{x}; \boldsymbol{\theta})$ maps any configuration $\boldsymbol{x}$ to a real number estimating the true probability $p_{\text{data}}(\boldsymbol{x})$.

- The maximum likelihood estimator for $\boldsymbol{\theta}$ is defined as

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} p_{\text{model}}(\mathbb{X}; \boldsymbol{\theta}), \tag{5.56}$$

$$= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}). \tag{5.57}$$

Maximum likelihood estimation

- This product over many probabilities can be inconvenient for various reasons. For example, it is prone to numerical underflow. To obtain a more convenient but equivalent optimization problem, we observe that taking the logarithm of the likelihood does not change its argmax but does conveniently transform a product into a sum:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\mathrm{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}). \tag{5.58}$$

- Because the argmax does not change when we rescale the cost function, we can divide by m to obtain a version of the criterion that is expressed as an expectation with respect to the empirical distribution $\hat{p}$ defined by the training data:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\mathrm{data}}} \log p_{\mathrm{model}}(\boldsymbol{x}; \boldsymbol{\theta}). \tag{5.59}$$

Maximum likelihood estimation

- One way to interpret maximum likelihood estimation is to view it as minimizing the dissimilarity between the empirical distribution $\hat{p}_{\text{data}}$, defined by the training set, and the model distribution, with the degree of dissimilarity between the two measured by the KL divergence.

- The KL divergence is given by

$$D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right) = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\left[\log \hat{p}_{\text{data}}(\boldsymbol{x}) - \log p_{\text{model}}(\boldsymbol{x})\right]. \tag{5.60}$$

- The term on the left is a function only of the data-generating process, not the model. This means when we train the model to minimize the KL divergence, we need only minimize

$$-\mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}}\left[\log p_{\text{model}}(\boldsymbol{x})\right], \tag{5.61}$$

Maximum likelihood estimation

- Minimizing this KL divergence corresponds exactly to minimizing the cross-entropy between the distributions.

- Many use the term "cross-entropy" to identify specifically the negative log-likelihood of a Bernoulli or $\mathrm{softmax}$ distribution. In fact, any loss consisting of a negative log-likelihood is a cross-entropy between the empirical distribution defined by the training set and the probability distribution defined by model. For example, mean squared error is the cross-entropy between the empirical distribution and a Gaussian model.

- Maximum likelihood is an attempt to make the model distribution match the empirical distribution $\hat{p}_{\mathrm{data}}$. Ideally, we want to match the true data-generating distribution p_{data}, but we have no direct access to this distribution.

Conditional log-likelihood

- The maximum likelihood estimator can readily be generalized to estimate a conditional probability $P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ in order to predict $\mathbf{y}$ given $\mathbf{x}$.

- If $\boldsymbol{X}$ represents all our inputs and $\boldsymbol{Y}$ all our observed targets, then the conditional maximum likelihood estimator is

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{Y} \mid \boldsymbol{X};\boldsymbol{\theta}). \tag{5.62}$$

- If the examples are assumed to be i.i.d., then this can be decomposed into

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log P(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)};\boldsymbol{\theta}). \tag{5.63}$$

Linear regression as maximum likelihood

Example: Linear Regression as Maximum Likelihood

- We now revisit linear regression from the point of view of maximum likelihood estimation. Instead of producing a single prediction $\hat{y}$, we now think of the model as producing a conditional distribution $p(y|\boldsymbol{x})$.

- We can imagine that with an infinitely large training set, we might see several training examples with the same input value $\boldsymbol{x}$ but different values of y. The goal of the learning algorithm is now to fit the distribution $p(y|\boldsymbol{x})$ to all those different y values that are all compatible with $\boldsymbol{x}$.

Linear regression as maximum likelihood

- Define $p(y|\boldsymbol{x}) = \mathcal{N}(y; \hat{y}(\boldsymbol{x}; \boldsymbol{\omega}), \sigma^2)$. The function $\hat{y}(\boldsymbol{x}; \boldsymbol{\omega})$ gives the prediction of the mean of the Gaussian. In this example, we assume that the variance is fixed to some constant σ^2 chosen by the user.

- Since the examples are assumed to be i.i.d., the conditional log-likelihood (equation 5.63) is given by

$$\sum_{i=1}^{m} \log p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \tag{5.64}$$

$$= -m \log \sigma - \frac{m}{2} \log(2\pi) - \sum_{i=1}^{m} \frac{\left\| \hat{y}^{(i)} - y^{(i)} \right\|^2}{2\sigma^2}, \tag{5.65}$$

where $\hat{y}^{(i)}$ is the output of the linear regression on the i-th input $\boldsymbol{x}^{(i)}$ and m is the number of the training examples.

Linear regression as maximum likelihood

- Comparing the log-likelihood with the mean squared error,

$$\sum_{i=1}^{m} \log p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \tag{5.64}$$

$$= -m \log \sigma - \frac{m}{2} \log(2\pi) - \sum_{i=1}^{m} \frac{\left\| \hat{y}^{(i)} - y^{(i)} \right\|^2}{2\sigma^2}, \tag{5.65}$$

  we immediately see that maximizing the log-likelihood with respect to $\boldsymbol{\omega}$ yields the same estimate of the parameters $\boldsymbol{\omega}$ as does minimizing the mean squared error.

- The two criteria have different values but the same location of the optimum. This justifies the use of the MSE as a maximum likelihood estimation procedure.

Consistency

- Usually we are also concerned with the behavior of an estimator as the amount of training data grows. In particular, we usually wish that, as the number of data points m in our dataset increases, our point estimates converge to the true value of the corresponding parameters. More formally, we would like that

$$\text{plim}_{m \to \infty} \hat{\theta}_m = \theta. \tag{5.55}$$

- The symbol plim indicates convergence in probability, meaning that for any $\epsilon > 0$, $P(|\hat{\theta}_m - \theta| > \epsilon) \to 0$ as $m \to \infty$. The condition described by equation 5.55 is known as consistency (sometimes referred to as weak consistency). Strong consistency refers to the almost sure convergence of $\hat{\theta}$ to θ. Almost sure convergence of a sequence of random variables $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots$ to a value $\boldsymbol{x}$ occurs when $p(\lim_{m \to \infty} \mathbf{x}^{(m)} = \boldsymbol{x}) = 1$.

Consistency

- Consistency ensures that the bias induced by the estimator diminishes as the number of data examples grows. However, the reverse is not true—asymptotic unbiasedness does not imply consistency.

- For example, consider estimating the mean parameter μ of a normal distribution $N(x; \mu, \sigma^2)$, with a dataset consisting of m samples: $\{x^{(1)}, \ldots, x^{(m)}\}$. We could use the first sample $x^{(1)}$, of the dataset as an unbiased estimator: $\hat{\theta} = x^{(1)}$. In that case, $\mathbb{E}(\hat{\theta}_m) = \theta$, so the estimator is unbiased no matter how many data points are seen. This, of course, implies that the estimate is asymptotically unbiased. However, this is not a consistent estimator as it is not the case that $\hat{\theta}_m \rightarrow \theta$ as $m \rightarrow \infty$.

Properties of maximum likelihood

- The maximum likelihood estimator can be shown to be the best estimator asymptotically, as the number of examples $m \to \infty$, in terms of its rate of convergence as m increases.

- The maximum likelihood estimator has the property of consistency under the following conditions:
  - The true distribution p_{data} must lie within the model family $p_{\text{model}}(\cdot; \boldsymbol{\theta})$. Otherwise, no estimator can recover p_{data}.
  - The true distribution p_{data} must correspond to exactly one value of $\boldsymbol{\theta}$. Otherwise, maximum likelihood can recover the correct p_{data} but will not be able to determine which value of $\boldsymbol{\theta}$ was used by the data-generating process.

Properties of maximum likelihood

- A way to measure how close we are to the true parameter is by the expected mean squared error, computing the squared difference between the estimated and true parameter values, where the expectation is over m training samples from the data-generating distribution.

- That parametric mean squared error decreases as m increases, and for m large, the Cramér-Rao lower bound shows that no consistent estimator has a lower MSE than the maximum likelihood estimator.

- For these reasons (consistency and efficiency), maximum likelihood is often considered the preferred estimator to use for machine learning.

Bayesian statistics

- The approach of frequentist statistics is based on estimating a single value of $\boldsymbol{\theta}$, then making all predictions thereafter based on that one estimate. Another approach, Bayesian statistics, is to consider all possible values of $\boldsymbol{\theta}$ when making a prediction.

- The frequentist perspective is that the true parameter value $\boldsymbol{\theta}$ is fixed but unknown, while the point estimate $\widehat{\boldsymbol{\theta}}$ is a random variable on account of it being a function of the dataset (which is seen as random).

- The Bayesian uses probability to reflect degrees of certainty in states of knowledge. The dataset is directly observed and so is not random. On the other hand, the true parameter $\boldsymbol{\theta}$ is unknown or uncertain and thus is represented as a random variable.

Bayesian statistics

- Before observing the data, we represent our knowledge of $\boldsymbol{\theta}$ using the prior probability distribution, $p(\boldsymbol{\theta})$ (sometimes referred to as simply "the prior").

- Generally, the machine learning practitioner selects a prior distribution that is quite broad (i.e., with high entropy) to reflect a high degree of uncertainty in the value of $\boldsymbol{\theta}$ before observing any data.

- For example, one might assume a priori that $\boldsymbol{\theta}$ lies in some finite range or volume, with a uniform distribution. Many priors instead reflect a preference for "simpler" solutions (such as smaller magnitude coefficients, or a function that is closer to being constant).

Bayesian statistics

- Now consider that we have a set of data samples $\{x^{(1)}, \ldots, x^{(m)}\}$. We can recover the effect of data on our belief about $\boldsymbol{\theta}$ by combining the data likelihood $p(x^{(1)}, \ldots, x^{(m)}|\boldsymbol{\theta})$ with the prior via Bayes' rule:

$$p(\boldsymbol{\theta} \mid x^{(1)}, \ldots, x^{(m)}) = \frac{p(x^{(1)}, \ldots, x^{(m)} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(x^{(1)}, \ldots, x^{(m)})} \tag{5.67}$$

- In the scenarios where Bayesian estimation is typically used, the prior begins as a relatively uniform or Gaussian distribution with high entropy, and the observation of the data usually causes the posterior to lose entropy and concentrate around a few highly likely values of the parameters.

Bayesian statistics

- Bayesian estimation offers two important differences. First, the Bayesian approach is to make predictions using a full distribution over $\boldsymbol{\theta}$. For example, after observing m examples, the predicted distribution over the next data sample, $x^{(m+1)}$, is given by

$$p(x^{(m+1)} \mid x^{(1)}, \ldots, x^{(m)}) = \int p(x^{(m+1)} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid x^{(1)}, \ldots, x^{(m)}) \, d\boldsymbol{\theta}. \qquad (5.68)$$

- After having observed $\{x^{(1)}, \ldots, x^{(m)}\}$, if we are still quite uncertain about the value of $\boldsymbol{\theta}$, then this uncertainty is incorporated directly into any predictions we might make.

- The Bayesian answer to the question of how to deal with the uncertainty in the estimator is to simply integrate over it, which tends to protect well against overfitting.

Bayesian statistics

- The second important difference is: The prior has an influence by shifting probability mass density towards regions of the parameter space that are preferred a priori. In practice, the prior often expresses a preference for models that are simpler or more smooth.

- Critics of the Bayesian approach identify the prior as a source of subjective human judgment.

- Bayesian methods typically generalize much better when limited training data is available but typically suffer from high computational cost when the number of training examples is large.

Bayesian statistics

Example: Bayesian Linear Regression

- Here we consider the Bayesian estimation approach to learning the linear regression parameters. In linear regression, we learn a linear mapping from an input vector $\boldsymbol{x} \in \mathbb{R}^n$ to predict the value of a scalar $y \in \mathbb{R}$. The prediction is parametrized by the vector $\boldsymbol{\omega} \in \mathbb{R}^n$:

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x}. \tag{5.69}$$

- Given a set of m training samples $(\boldsymbol{X}^{(\text{train})}, \boldsymbol{y}^{(\text{train})})$, we can express the prediction of y over the entire training

$$\hat{\boldsymbol{y}}^{(\text{train})} = \boldsymbol{X}^{(\text{train})} \boldsymbol{w}. \tag{5.70}$$

Bayesian statistics

- Expressed as a Gaussian conditional distribution on y, we have

$$p(\boldsymbol{y}^{(\text{train})} \mid \boldsymbol{X}^{(\text{train})}, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}^{(\text{train})}; \boldsymbol{X}^{(\text{train})}\boldsymbol{w}, \boldsymbol{I}) \tag{5.71}$$

$$\propto \exp\left(-\frac{1}{2}(\boldsymbol{y}^{(\text{train})} - \boldsymbol{X}^{(\text{train})}\boldsymbol{w})^{\top}(\boldsymbol{y}^{(\text{train})} - \boldsymbol{X}^{(\text{train})}\boldsymbol{w})\right), \tag{5.72}$$

  where we follow the standard MSE formulation in assuming that the Gaussian variance on y is one.

- In what follows, to reduce the notational burden, we refer to $(\boldsymbol{X}^{(\text{train})}, \boldsymbol{y}^{(\text{train})})$ as simply $(\boldsymbol{X}, \boldsymbol{y})$.

Bayesian statistics

- To determine the posterior distribution over the model parameter vector $\boldsymbol{\omega}$, we first need to specify a prior distribution. The prior should reflect our naive belief about the value of these parameters. While it is sometimes difficult or unnatural to express our prior beliefs in terms of the parameters of the model, in practice we typically assume a fairly broad distribution, expressing a high degree of uncertainty about $\boldsymbol{\theta}$.

- For real-valued parameters it is common to use a Gaussian as a prior distribution,

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0) \propto \exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Lambda}_0^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_0)\right), \tag{5.73}$$

where $\boldsymbol{\mu}_0$ and $\boldsymbol{\Lambda}_0$ are the prior distribution mean vector and covariance matrix respectively.

Bayesian statistics

- With the prior thus specified, we can now proceed in determining the posterior distribution over the model parameters:

$$p(\boldsymbol{w} \mid \boldsymbol{X}, \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w}) \tag{5.74}$$

$$\propto \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\top}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})\right) \exp\left(-\frac{1}{2}(\boldsymbol{w}-\boldsymbol{\mu}_0)^{\top}\boldsymbol{\Lambda}_0^{-1}(\boldsymbol{w}-\boldsymbol{\mu}_0)\right) \tag{5.75}$$

$$\propto \exp\left(-\frac{1}{2}\left(-2\boldsymbol{y}^{\top}\boldsymbol{X}\boldsymbol{w}+\boldsymbol{w}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{w}+\boldsymbol{w}^{\top}\boldsymbol{\Lambda}_0^{-1}\boldsymbol{w}-2\boldsymbol{\mu}_0^{\top}\boldsymbol{\Lambda}_0^{-1}\boldsymbol{w}\right)\right). \tag{5.76}$$

Bayesian statistics

- We now define $\boldsymbol{\Lambda}_m = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{\Lambda}_0^{-1})^{-1}$ and $\boldsymbol{\mu}_m = \boldsymbol{\Lambda}_m(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\Lambda}_0^{-1}\boldsymbol{\mu}_0)$. Using these new variables, we find that the posterior may be rewritten as a Gaussian distribution:

$$p(\boldsymbol{w} \mid \boldsymbol{X}, \boldsymbol{y}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu}_m)^{\top}\boldsymbol{\Lambda}_m^{-1}(\boldsymbol{w} - \boldsymbol{\mu}_m) + \frac{1}{2}\boldsymbol{\mu}_m^{\top}\boldsymbol{\Lambda}_m^{-1}\boldsymbol{\mu}_m\right) \tag{5.77}$$

$$\propto \exp\left(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu}_m)^{\top}\boldsymbol{\Lambda}_m^{-1}(\boldsymbol{w} - \boldsymbol{\mu}_m)\right). \tag{5.78}$$

- All terms that do not include the parameter vector $\boldsymbol{\omega}$ have been omitted; they are implied by the fact that the distribution must be normalized to integrate to 1.

Bayesian statistics

- In most situations, we set $\boldsymbol{\mu}_0$ to $\mathbf{0}$. If we set $\boldsymbol{\Lambda}_0 = \frac{1}{\alpha}\boldsymbol{I}$, then $\boldsymbol{\mu}_m$ gives the same estimate of $\boldsymbol{\omega}$ as does frequentist linear regression with a weight decay penalty of $\alpha\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}$.

- One difference is that the Bayesian estimate is undefined if α is set to zero—we are not allowed to begin the Bayesian learning process with an infinitely wide prior on $\boldsymbol{\omega}$.

- The more important difference is that the Bayesian estimate provides a covariance matrix, showing how likely all the different values of $\boldsymbol{\omega}$ are, rather than providing only the estimate $\boldsymbol{\mu}_m$.

MAP estimation

- While the most principled approach is to make predictions using the full Bayesian posterior distribution over the parameter $\boldsymbol{\theta}$, it is still often desirable to have a single point estimate.

- One common reason for desiring a point estimate is that most operations involving the Bayesian posterior for most interesting models are intractable, and a point estimate offers a tractable approximation.

MAP estimation

- Rather than simply returning to the maximum likelihood estimate, we can still gain some of the benefit of the Bayesian approach by allowing the prior to influence the choice of the point estimate.

- One rational way to do this is to choose the maximum a posteriori (MAP) point estimate. The MAP estimate chooses the point of maximal posterior probability (or maximal probability density in the more common case of continuous $\boldsymbol{\theta}$):

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{x} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}). \tag{5.79}$$

- We recognize, on the righthand side, $\log p(x|\boldsymbol{\theta})$, that is, the standard log-likelihood term, and $\log p(\boldsymbol{\theta})$, corresponding to the prior distribution.

MAP estimation

- As an example, consider a linear regression model with a Gaussian prior on the weights $\boldsymbol{\omega}$. If this prior is given by $\mathcal{N}(\boldsymbol{\omega}; \mathbf{0}, \frac{1}{\lambda}\boldsymbol{I}^2)$, then the log-prior term in equation 5.79 is proportional to the familiar $\lambda\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}$ weight decay penalty, plus a term that does not depend on $\boldsymbol{\omega}$ and does not affect the learning process. MAP Bayesian inference with a Gaussian prior on the weights thus corresponds to weight decay.

- As with full Bayesian inference, MAP Bayesian inference has the advantage of leveraging information that is brought by the prior and cannot be found in the training data. This additional information helps to reduce the variance in the MAP point estimate. However, it does so at the price of increased bias.

MAP estimation

- Not all regularization penalties correspond to MAP Bayesian inference. Some regularizer terms may not be the logarithm of a probability distribution. Other regularization terms depend on the data, which of course a prior probability distribution is not allowed to do.

- MAP Bayesian inference provides a straightforward way to design complicated yet interpretable regularization terms. For example, a more complicated penalty term can be derived by using a mixture of Gaussians, rather than a single Gaussian distribution, as the prior.

Challenges motivating deep learning

- The simple machine learning algorithms described in this chapter work well on a wide variety of important problems. They have not succeeded, however, in solving the central problems in AI, such as recognizing speech or recognizing objects.

- The development of deep learning was motivated in part by the failure of traditional algorithms to generalize well on such AI tasks.

The curse of dimensionality

- Many machine learning problems become exceedingly difficult when the number of dimensions in the data is high. This phenomenon is known as the curse of dimensionality.

- Of particular concern is that the number of possible distinct configurations of a set of variables increases exponentially as the number of variables increases.

The curse of dimensionality

- One challenge posed by the curse of dimensionality is a statistical challenge. A statistical challenge arises because the number of possible configurations of $\boldsymbol{x}$ is much larger than the number of training examples.

- Let us consider that the input space is organized into a grid. We can describe low-dimensional space with a small number of grid cells that are mostly occupied by the data. When generalizing to a new data point, we can usually tell what to do simply by inspecting the training examples that lie in the same cell as the new input.

The curse of dimensionality

The curse of dimensionality

- For example, if estimating the probability density at some point $\boldsymbol{x}$, we can just return the number of training examples in the same unit volume cell as $\boldsymbol{x}$, divided by the total number of training examples.

- If we wish to classify an example, we can return the most common class of training examples in the same cell.

- If we are doing regression, we can average the target values observed over the examples in that cell.

The curse of dimensionality

- But what about the cells for which we have seen no example?

- In high-dimensional spaces, the number of configurations is huge, much larger than our number of examples, a typical grid cell has no training example associated with it. How could we possibly say something meaningful about these new configurations?

- Many traditional machine learning algorithms simply assume that the output at a new point should be approximately the same as the output at the nearest training point.

Local constancy and smoothness regularization

- To generalize well, machine learning algorithms need to be guided by the prior.
- Among the most widely used of these implicit “priors” is the smoothness prior, or local constancy prior. This prior states that the function we learn should not change very much within a small region.
- Many simpler algorithms rely exclusively on this prior to generalize well, and as a result, they fail to scale to the statistical challenges involved in solving AI-level tasks. The smoothness prior alone is insufficient for these tasks.

Local constancy and smoothness regularization

- The core idea in deep learning is that we assume that the data was generated by the composition of factors, or features, potentially at multiple levels in a hierarchy.

- These apparently mild assumptions allow an exponential gain in the relationship between the number of examples and the number of regions that can be distinguished. The exponential advantages conferred by the use of deep distributed representations counter the exponential challenges posed by the curse of dimensionality.

Probabilistic supervised learning

- Most supervised learning algorithms are based on estimating a probability distribution $p(y|\boldsymbol{x})$. We can do this simply by using maximum likelihood estimation to find the best parameter vector $\boldsymbol{\theta}$ for a parametric family of distributions $p(y|\boldsymbol{x};\boldsymbol{\theta})$.

- We can generalize linear regression to the classification scenario by defining a different family of probability distributions. If we have two classes, class 0 and class 1, then we need only specify the probability of one of these classes. The probability of class 1 determines the probability of class 0, because these two values must add up to 1.

Probabilistic supervised learning

- The mean of a distribution over a binary variable must always be between 0 and 1. One way to solve this problem is to use the logistic sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ to squash the output of the linear function into the interval (0, 1) and interpret that value as a probability:

$$p(y = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}). \tag{5.81}$$

- This approach is known as logistic regression (a somewhat strange name since we use the model for classification rather than regression).

Probabilistic supervised learning

- In the case of linear regression, we were able to find the optimal weights by solving the normal equations. Logistic regression is somewhat more difficult. There is no closed-form solution for its optimal weights. Instead, we must search for them by maximizing the log-likelihood. We can do this by minimizing the negative log-likelihood using gradient descent.

- This same strategy can be applied to essentially any supervised learning problem, by writing down a parametric family of conditional probability distributions over the right kind of input and output variables.

Generative vs. discriminative models

- We can identify three distinct approaches to solving decision problems. These are given, in decreasing order of complexity, by:

(a) First solve the inference problem of determining the class-conditional densities $p(\mathbf{x}|C_k)$ for each class C_k individually. Also separately infer the prior class probabilities $p(C_k)$. Then use Bayes' theorem in the form

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} \tag{1.82}$$

to find the posterior class probabilities $p(C_k|\mathbf{x})$. As usual, the denominator in Bayes' theorem can be found in terms of the quantities appearing in the numerator, because

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|C_k)p(C_k). \tag{1.83}$$

Generative vs. discriminative models

Equivalently, we can model the joint distribution $p(\mathbf{x}, C_k)$ directly and then normalize to obtain the posterior probabilities.

Having found the posterior probabilities, we use decision theory to determine class membership for each new input $\mathbf{x}$.

Approaches that explicitly or implicitly model the distribution of inputs as well as outputs are known as generative models, because by sampling from them it is possible to generate synthetic data points in the input space.

Generative vs. discriminative models

(b) First solve the inference problem of determining the posterior class probabilities $p(C_k|\mathbf{x})$, and then subsequently use decision theory to assign each new $\mathbf{x}$ to one of the classes. Approaches that model the posterior probabilities directly are called discriminative models.

Generative vs. discriminative models

(c) Find a function $f(\mathbf{x})$, called a discriminant function, which maps each input $\mathbf{x}$ directly onto a class label. For instance, in the case of two-class problems, $f(\cdot)$ might be binary valued and such that $f = 0$ represents class C_1 and $f = 1$ represents class C_2. In this case, probabilities play no role.

Generative vs. discriminative models

- The generative model approach is the most demanding because it involves finding the joint distribution over both $\mathbf{x}$ and C_k. For many applications, $\mathbf{x}$ will have high dimensionality, and consequently we may need a large training set in order to be able to determine the class-conditional densities to reasonable accuracy.

- One advantage of the generative model approach is that it also allows the marginal density of data $p(\mathbf{x})$ to be determined from

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k). \qquad (1.83)$$

This can be useful for detecting new data points that have low probability under the model and for which the predictions may be of low accuracy, which is known as outlier detection or novelty detection.

Generative vs. discriminative models

- However, if we only wish to make classification decisions, then it can be wasteful of computational resources, and excessively demanding of data, to find the joint distribution $p(\mathbf{x}, C_k)$ when in fact we only really need the posterior probabilities $p(C_k|\mathbf{x})$, which can be obtained directly through the discriminative model approach.